

frrouting — CWE-407 Disclosure Brief

FRRouting — CWE-407 Disclosure Brief

2026-03-26 · Confidential pre-disclosure

Finding

Two $O(n^2)$ defects in FRRouting's OSPF implementation. One patched in a recent commit (frrouting-0001). One unpatched and active on every topology change event (frrouting-0002).

The Defect

frrouting-0001 (PATCHED): `ospf_ti_lfa.c` lines 72, 114, 227, 278, 285 — five separate `listnode_lookup()` calls inside the TI-LFA backup path algorithm. Each is a linear scan of a linked list. Five scans per path computation, inside a function called for every protected prefix.

frrouting-0002 (UNPATCHED): `ospf_spf.c:275` — `listnode_lookup(vp->parent->children, v)` inside Dijkstra's main SPF loop.

This is the more serious defect. It fires on every OSPF SPF computation — meaning every link flap, router restart, or metric change on every FRRouting router running OSPF.

Complexity Proof

Dijkstra's SPF loop processes each vertex `v`. For each `v`, it walks `v->parent->children` from the head to find `v`. The children list grows as Dijkstra processes vertices. On a hub-and-spoke topology with H spoke routers:

- The hub's children list accumulates H entries as spokes are processed
- Each subsequent spoke triggers a scan of that list to locate its predecessor
- Scan lengths: 1, 2, 3, ... $H \rightarrow \text{sum} = H(H+1)/2 = O(H^2)$

For a WAN aggregation hub with 100 spokes: ~5,000 extra list operations per SPF run. For 200 spokes: ~20,000. Every topology event — link flap, BFD timeout, metric change — triggers a full SPF recomputation. On busy ISP cores this fires continuously.

The standard Dijkstra implementation does not require this lookup at all. It is an artifact of the linked-list data structure choice for `children`.

Impact

Every FRRouting router running OSPF on a hub-and-spoke or partial-mesh topology. Affected deployments include:

- ISP hub sites aggregating spoke CPE or PE routers
- Enterprise MPLS cores with hub-site route reflectors
- Carrier peering fabrics with high-degree OSPF speakers
- Data center spine/leaf fabrics running OSPF for underlay

The $O(H^2)$ cost is paid on every SPF computation. In environments with frequent topology events (BFD sub-second timers, unstable links), SPF CPU consumption is materially higher than it needs to be. Under load, this delays convergence — the opposite of what OSPF is supposed to provide.

The Fix

Replace `struct list *children` with a hash-based lookup structure. FRRouting already uses `hash_*` APIs extensively elsewhere in the codebase — the pattern is established.

Minimal fix: add a companion `struct hash *children_hash` to the SPF vertex struct. Populate it alongside the list. Replace the `listnode_lookup` call with `hash_lookup`. The list can be retained for ordered traversal; the hash is used only for membership testing.

Alternative: the `OSPF_SPF_RESULT_PARENT` tracking already partially in scope could be extended to carry an O(1) membership flag, avoiding the lookup entirely.

The TI-LFA fix (frrouting-0001) already demonstrates the team knows how to address this class of defect. frrouting-0002 is the same pattern in a hotter code path.

What We Ask

1. Confirm receipt and assign a tracker reference.
2. Assess whether frrouting-0002 warrants a CVE (CWE-407, algorithmic complexity, degraded convergence under load).
3. Coordinate a disclosure date — we are targeting 90 days from first contact.
4. We will credit FRRouting in the public disclosure. If you have a preferred acknowledgment format, let us know.

Contact: see cover email. This brief is confidential until coordinated disclosure.